

Attorney Docket No. 06618-303001
Application No. 09/295,957
Amendment dated May 20, 2004
Reply to Office Action dated January 20, 2004

Amendment to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Previously Presented) A distributed system framework for a networked environment, including a plurality of process objects, each of said process objects including:

(a) a program method for creating at least one inbox for storing messages received from another process object;

(b) a program method for creating at least one outbox for storing messages to be transmitted to another process object;

(c) a freeze method that saves the state of the process object to persistent storage, thereby changing the process object to a frozen process object which does not use operating system resources;

(d) a thaw method that restores the frozen process object from the persistent storage, thereby changing the frozen process object to a ready process object;

(e) a program method for interconnecting each created outbox of the process object to a created inbox of at least one other process object, thereby establishing a personal network between the process object and such other process objects within

BEST AVAILABLE COPY

Attorney Docket No. 06618-303001
Application No. 09/295,957
Amendment dated May 20, 2004
Reply to Office Action dated January 20, 2004

a communication session to perform at least one task by passing messages between the interconnected outboxes and inboxes.

2. (Original) The distributed system framework of claim 1, each process object further including at least one mail daemon object, for controlling the order of messages in each inbox.

3. (Previously Presented) The distributed system framework, of claim 1, each process object further including a summoning response method, for instantiating the process object if the process object is summoned by another process object.

4. (Original) The distributed system framework of claim 3, wherein the summoning response method causes the thaw method of the process object to be invoked if the process object is frozen when summoned by another process.

5. (Original) The distributed system framework of claim 1, wherein each message includes a snapshot variable that indicates whether a process object has recorded its state.

6. (Currently Amended) ~~The distributed system framework of claim 5,~~ A distributed system framework for a networked

Attorney Docket No. 06618-303001
Application No. 09/295,957
Amendment dated May 20, 2004
Reply to Office Action dated January 20, 2004

environment, including a plurality of process objects, each of said process objects including:

(a) a program method for creating at least one inbox for storing messages received from another process object;

(b) a program method for creating at least one outbox for storing messages to be transmitted to another process object;

(c) a freeze method that saves the state of the process object to persistent storage, thereby changing the process object to a frozen process object which does not use operating system resources;

(d) a thaw method that restores the frozen process object from the persistent storage, thereby changing the frozen process object to a ready process object;

(e) a program method for interconnecting each created outbox of the process object to a created inbox of at least one other process object, thereby establishing a personal network between the process object and such other process objects within a communication session to perform at least one task by passing messages between the interconnected outboxes and inboxes;

wherein each message includes a snapshot variable that indicates whether a process object has recorded its state; and

wherein the snapshot variable is a date field, and each process object includes a snapshot method that saves the state

Attorney Docket No. 06618-303001
Application No. 09/295,957
Amendment dated May 20, 2004
Reply to Office Action dated January 20, 2004

of the process object if a date field value within a received message is later ~~that~~ than current date value for the process object, updates the current date value for the process object to the date field value of the received message, and increments a clock for the process object to a value exceeding the date field value of the received message.

BEST AVAILABLE COPY